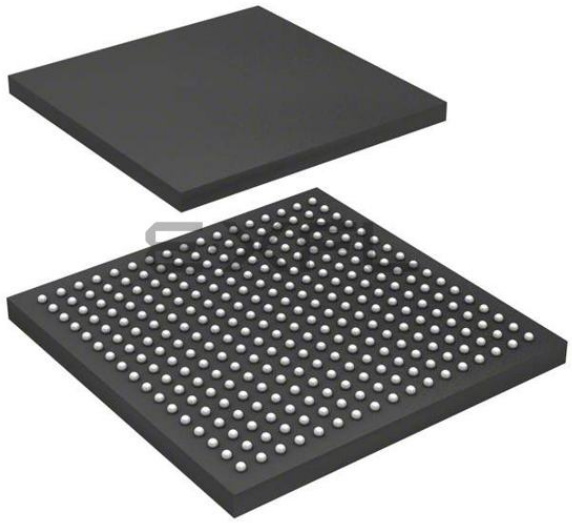


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	500MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR1, LPDDR2, LPDDR3, DDR2, DDR3, DDR3L, QSPI
Graphics Acceleration	Yes
Display & Interface Controllers	Keyboard, LCD, Touchscreen
Ethernet	10/100Mbps (1)
SATA	-
USB	USB 2.0 + HSIC
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 105°C (TA)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC
Package / Case	289-LFBGA
Supplier Device Package	289-LFBGA (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d27c-cn

Table 6-2: Pin Description (Continued)

289-pin BGA	256-pin BGA	196-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO peripheral				Reset State (Signal, Dir, PU, PD, HiZ, ST)(1)(2)
					Signal	Dir	Signal	Dir	Func	Signal	Dir	IO Set	
K7	K4	H5	VDDANA	GPIO_AD	PD15	I/O	PTC_Y4	–	A	TDI	I	1	PIO, I, PU, ST
									B	FLEXCOM4_IO3	O	2	
									C	UTMI_CDRCPDIVEN	I	1	
									D	GTX0	O	2	
									E	ISC_PCK	I	2	
									F	ISC_D7	I	4	
L1	K1	G1	VDDANA	GPIO_AD	PD16	I/O	PTC_Y5	–	A	TDO	O	1	PIO, I, PU, ST
									B	FLEXCOM4_IO4	O	2	
									C	UTMI_CDRBISTEN	I	1	
									D	GTX1	O	2	
									E	ISC_VSYNC	I	2	
									F	ISC_D8	I	4	
K2	K2	G2	VDDANA	GPIO_AD	PD17	I/O	PTC_Y6	–	A	TMS	I	1	A, PU, ST
									C	UTMI_CDRCPSELDIV	O	1	
									D	GMDC	O	2	
									E	ISC_HSYNC	I	2	
									F	ISC_D9	I	4	
J5	L5	G3	VDDANA	GPIO_AD	PD18	I/O	PTC_Y7	–	A	NTRST	I	1	PIO, I, PU, ST
									D	GMDIO	I/O	2	
									E	ISC_FIELD	I	2	
									F	ISC_D10	I	4	
K6	L4	H4	VDDANA	GPIO_AD	PD19	I/O	AD0	–	A	PCK0	O	1	PIO, I, PU, ST
									B	TWD1	I/O	3	
									C	URXD2	I	3	
									E	I2SC0_CK	I/O	2	
									F	ISC_D11	I	4	
M2	M1	J1	VDDANA	GPIO_AD	PD20	I/O	AD1	–	A	TIOA2	I/O	3	PIO, I, PU, ST
									B	TWCK1	I/O	3	
									C	UTXD2	O	3	
									E	I2SC0_MCK	O	2	
									F	ISC_PCK	I	4	
N1	M2	K1	VDDANA	GPIO_AD	PD21	I/O	AD2	–	A	TIOB2	I/O	3	PIO, I, PU, ST
									B	TWD0	I/O	4	
									C	FLEXCOM4_IO0	I/O	3	
									E	I2SC0_WS	I/O	2	
									F	ISC_VSYNC	I	4	

Table 6-3: Pin Description (SAMA5D23 pins different from those in Table 6-2 “Pin Description”)

196-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO peripheral				Reset State (Signal, Dir, PU, PD, HiZ, ST)(1)(2)
			Signal	Dir	Signal	Dir	Func	Signal	Dir	IO Set	
M12	VDDIOP1	GPIO	PC0	I/O	–	–	A	LCDDAT21	O	1	PIO, I, PU, ST
							B	A23	O	1	
							C	FLEXCOM0_IO4	O	1	
							D	TWCK0	I/O	1	
							F	ISC_D6	I	3	
M13	VDDIOP1	GPIO	PC1	I/O	–	–	A	LCDDAT22	O	1	PIO, I, PU, ST
							B	A24	O	1	
							C	CANTX0	O	1	
							D	SPI1_SPCK	I/O	1	
							E	I2SC0_CK	I/O	1	
							F	ISC_D7	I	3	
L4	VDDBU	–	PIOBU1	–	–	–	–	–	–	–	–
L3	VDDBU	–	PIOBU2	–	–	–	–	–	–	–	–
M5	VDDBU	–	PIOBU3	–	–	–	–	–	–	–	–
L6	VDDBU	–	PIOBU5	–	–	–	–	–	–	–	–
P13	VDDFUSE	power	VDDFUSE	I	–	–	–	–	–	–	–

Note 1: Signal = ‘PIO’ if GPIO; Dir = Direction; PB/CU = Pull-up; PD = Pull-down; HiZ = High impedance; ST = Schmitt Trigger

2: The GPIOs’ reset state is not guaranteed during the powerup phase. During this phase, the GPIOs are in input pullup mode and they take their reset value only after VDDCORE POR reset has been released. If a GPIO must be at level zero at powerup, it is recommended to connect an external pulldown to guarantee this state.

The SAMA5D28B/C are not pin-to-pin compatible with SAMA5D28A, SAMA5D26A/B/C and SAMA5D27A/B/C. Table 6-4 provides the differences in pinout.

Table 6-4: Pin Description (SAMA5D28B/C pins different from those in Table 6-2 “Pin Description”)

289-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO peripheral				Reset State (Signal, Dir, PU, PD, HiZ, ST)(1)(2)
			Signal	Dir	Signal	Dir	Func	Signal	Dir	IO Set	
P4	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
N5	GNDCORE	ground	GNDCORE	I	–	–	–	–	–	–	–
R2	VDDBU	–	WKUP	–	–	–	–	–	–	–	–
N6	VDDBU	–	PIOBU0	–	–	–	–	–	–	–	–
M8	VDDBU	–	PIOBU2	–	–	–	–	–	–	–	–
P6	VDDBU	–	PIOBU3	–	–	–	–	–	–	–	–
P5	VDDBU	–	PIOBU4	–	–	–	–	–	–	–	–
R5	VDDBU	–	PIOBU5	–	–	–	–	–	–	–	–
N7	VDDBU	–	PIOBU6	–	–	–	–	–	–	–	–
M5	VDDBU	–	PIOBU7	–	–	–	–	–	–	–	–
R3	VDDBU	power	VDDBU	I	–	–	–	–	–	–	–
R4	GNDBU	ground	GNDBU	I	–	–	–	–	–	–	–

8.1 Embedded Memories

8.1.1 Internal SRAM

The SAMA5D2 embeds a total of 128 Kbytes of high-speed SRAM. After reset, and until the Remap command is performed, the SRAM is accessible at address 0x0020 0000. When the AXI Bus Matrix is remapped, the SRAM is also available at address 0x0.

The device features a second 128-Kbyte SRAM that can be allocated either to the L2 cache controller or used as an internal SRAM. After reset, this block is connected to the system SRAM, making the two 128-Kbyte RAMs contiguous. The SRAM_SEL bit, located in the SFR_L2CC_HRAMC register, is used to reassign this memory as a L2 cache memory.

8.1.2 Internal ROM

The product embeds one 160-Kbyte secured internal ROM mapped at address 0 after reset. The ROM contains a standard and secure bootloader as well as the BCH (Bose, Chaudhuri and Hocquenghem) code tables for NAND Flash ECC correction. The memory area containing the secure boot is automatically hidden after the execution of the secure boot while the one containing the code tables for ECC remains visible.

8.1.3 Boot Strategies

For standard boot strategies, refer to Section 16. “Standard Boot Strategies” of this datasheet.

For secure boot strategies, refer to the document “SAMA5D2x Secure Boot Strategy”, document no. 44040 (Non-Disclosure Agreement required).

8.2 External Memory

The SAMA5D2 offers connections to a wide range of external memories or to parallel peripherals.

8.2.1 External Bus Interface

The External Bus Interface (EBI) is a 16-bit wide interface working at MCK/2.

The EBI supports:

- Static memories
- 8-bit NAND Flash with 32-bit BCH ECC
- 16-bit NAND Flash

EBI I/Os accept three drive levels (Low, Medium, High) to avoid overshoots and provide the best performances according to the bus load and external memories voltage.

The drive levels are configured with the DRVSTR field in the PIO Configuration Register (PIO_CFGRx) if the corresponding line is non-secure or the Secure PIO Configuration Register (S_PIO_CFGRx) if the I/O line is secure.

At reset, the selected drive is low. The user must make sure to program the correct drive according to the device load. The I/O embeds serial resistors for impedance matching.

8.2.2 Supported Memories on DDR2/DDR3/LPDDR1/LPDDR2/LPDDR3 Interface

- 16-bit or 32-bit external interface
- 512 Mbytes of address space on DDR CS and DDR/AES CS in 32-bit mode
- 256 Mbytes of address space on DDR CS and DDR/AES CS in 16-bit mode
- Supports 16-bit or 32-bit 8-bank DDR2, DDR3, LPDDR1, LPDDR2 and LPDDR3 memories
- Automatic drive level control
- Multiport
- Scramblable data path
- Port 0 of this interface has an embedded automatic AES encryption and decryption mechanism (refer to Section 59. “Advanced Encryption Standard Bridge (AESB)”). Writing to or reading from the address 0x40000000 may trigger the encryption and decryption mechanism depending on the AESB on External Memories configuration.
- TrustZone: The multiport feature of this interface implies TrustZone configuration constraints. Refer to Section 18.12 “TrustZone Extension to AHB and APB” for more details.

8.2.3 Supported Memories on Static Memories and NAND Flash Interfaces

The Static Memory Controller is dedicated to interfacing external memory devices:

- Asynchronous SRAM-like memories and parallel peripherals

36.7 AHB Multiport DDR-SDRAM Controller (MPDDRC) User Interface

The User Interface is connected to the APB bus. The MPDDRC is programmed using the registers listed in Table 36-31.

Table 36-31: Register Mapping

Offset	Register	Name	Access	Reset
0x00	Mode Register	MPDDRC_MR	Read/Write	0x00000000
0x04	Refresh Timer Register	MPDDRC_RTR	Read/Write	0x03000000
0x08	Configuration Register	MPDDRC_CR	Read/Write	0x00207024
0x0C	Timing Parameter 0 Register	MPDDRC_TPR0	Read/Write	0x20227225
0x10	Timing Parameter 1 Register	MPDDRC_TPR1	Read/Write	0x3C80808
0x14	Timing Parameter 2 Register	MPDDRC_TPR2	Read/Write	0x00042062
0x18	Reserved	–	–	–
0x1C	Low-Power Register	MPDDRC_LPR	Read/Write	0x00010000
0x20	Memory Device Register	MPDDRC_MD	Read/Write	0x13
0x24	Reserved	–	–	–
0x28	Low-power DDR2 Low-power DDR3 Low-power Register	MPDDRC_LPDDR23_LPR	Read/Write	0x00000000
0x2C	Low-power DDR2 Low-power DDR3 and DDR3 Calibration and MR4 Register	MPDDRC_LPDDR2_LPDDR3_DDR3_CAL_MR4	Read/Write	0x00000000
0x30	Low-power DDR2 Low-power DDR3 and DDR3 Timing Calibration Register	MPDDRC_LPDDR2_LPDDR3_DDR3_TIM_CAL	Read/Write	0x06
0x34	I/O Calibration Register	MPDDRC_IO_CALIBR	Read/Write	0x00870000
0x38	OCMS Register	MPDDRC_OCMS	Read/Write	0x00000000
0x3C	OCMS KEY1 Register	MPDDRC_OCMS_KEY1	Write-only	–
0x40	OCMS KEY2 Register	MPDDRC_OCMS_KEY2	Write-only	–
0x44	Configuration Arbiter Register	MPDDRC_CONF_ARBITER	Read/Write	0x00000000
0x48	Timeout Register	MPDDRC_TIMEOUT	Read/Write	0x00000000
0x4C	Request Port 0-1-2-3 Register	MPDDRC_REQ_PORT_0123	Read/Write	0x00000000
0x50	Request Port 4-5-6-7 Register	MPDDRC_REQ_PORT_4567	Read/Write	0x00000000
0x54	Current/Maximum Bandwidth Port 0-1-2-3 Register	MPDDRC_BDW_PORT_0123	Read-only	0x00000000
0x58	Current/Maximum Bandwidth Port 4-5-6-7 Register	MPDDRC_BDW_PORT_4567	Read-only	0x00000000
0x5C	Read Data Path Register	MPDDRC_RD_DATA_PATH	Read/Write	0x00000000
0x60	Monitor Configuration Register	MPDDRC_MCFGR	Read/Write	0x00000000
0x64	Monitor Address High/Low Port 0 Register	MPDDRC_MADDR0	Read/Write	0x00000000
0x68	Monitor Address High/Low Port 1 Register	MPDDRC_MADDR1	Read/Write	0x00000000
0x6C	Monitor Address High/Low Port 2 Register	MPDDRC_MADDR2	Read/Write	0x00000000
0x70	Monitor Address High/Low Port 3 Register	MPDDRC_MADDR3	Read/Write	0x00000000
0x74	Monitor Address High/Low Port 4 Register	MPDDRC_MADDR4	Read/Write	0x00000000
0x78	Monitor Address High/Low Port 5 Register	MPDDRC_MADDR5	Read/Write	0x00000000

Table 38-2: DMA Channels Definition (XDMAC0) (Continued)

Instance Name	Channel T/R	Interface Number	XDMAC_CCx.CSIZE Required Value
FLEXCOM3	Transmit	17	0
FLEXCOM3	Receive	18	
FLEXCOM4	Transmit	19	0
FLEXCOM4	Receive	20	
SSC0	Transmit	21	0
SSC0	Receive	22	
SSC1	Transmit	23	0
SSC1	Receive	24	
ADC	Receive	25	0
AES	Transmit	26	0 or 2 (see AES Section 60.4.4.3 “DMA Mode”)
AES	Receive	27	
TDES	Transmit	28	0
TDES	Receive	29	
SHA	Transmit	30	4
I2SC0	Transmit	31	0
I2SC0	Receive	32	
I2SC1	Transmit	33	0
I2SC1	Receive	34	
UART0	Transmit	35	0
UART0	Receive	36	
UART1	Transmit	37	0
UART1	Receive	38	
UART2	Transmit	39	0
UART2	Receive	40	
UART3	Transmit	41	0
UART3	Receive	42	
UART4	Transmit	43	0
UART4	Receive	44	
TC0	Receive	45	0
TC1	Receive	46	
CLASSD	Transmit	47	0
QSPI1	Transmit	48	0
QSPI1	Receive	49	
PDMIC	Receive	50	0

39.7.66 Overlay 2 Configuration Register 0

Name: LCDC_OVR2CFG0

Address: 0xF000026C

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	LOCKDIS	ROTDIS	–	–	–	DLBO

7	6	5	4	3	2	1	0
–	–	BLEN		–	–	–	–

BLEN: AHB Burst Length

Value	Name	Description
0	AHB_SINGLE	AHB Access is started as soon as there is enough space in the FIFO to store one data. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.
1	AHB_INCR4	AHB Access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. An AHB INCR4 Burst is used. SINGLE, INCR and INCR4 bursts are used. INCR is used for a burst of 2 and 3 beats.
2	AHB_INCR8	AHB Access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. An AHB INCR8 Burst is used. SINGLE, INCR, INCR4 and INCR8 bursts are used. INCR is used for a burst of 2 and 3 beats.
3	AHB_INCR16	AHB Access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. An AHB INCR16 Burst is used. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.

DLBO: Defined Length Burst Only For Channel Bus Transaction

0: Undefined length INCR burst is used for 2 and 3 beats burst.

1: Only Defined Length burst is used (SINGLE, INCR4, INCR8 and INCR16).

ROTDIS: Hardware Rotation Optimization Disable

0: Rotation optimization is enabled.

1: Rotation optimization is disabled.

LOCKDIS: Hardware Rotation Lock Disable

0: AHB lock signal is asserted when a rotation is performed.

1: AHB lock signal is cleared when a rotation is performed.

39.7.127 High-End Overlay Configuration Register 32

Name: LCDC_HEOCFG32

Address: 0xF000040C

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
XPHI7COEFF4							

XPHI7COEFF4: Horizontal Coefficient for phase 7 tap 4

Coefficient format is 1 sign bit and 7 fractional bits.

39.7.131 High-End Overlay Configuration Register 36

Name: LCDC_HEOCFG36

Address: 0xF000041C

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
YPHI3COEFF2							

15	14	13	12	11	10	9	8
YPHI3COEFF1							

7	6	5	4	3	2	1	0
YPHI3COEFF0							

YPHI3COEFF0: Vertical Coefficient for phase 3 tap 0

Coefficient format is 1 sign bit and 7 fractional bits.

YPHI3COEFF1: Vertical Coefficient for phase 3 tap 1

Coefficient format is 1 magnitude bit and 7 fractional bits.

YPHI3COEFF2: Vertical Coefficient for phase 3 tap 2

Coefficient format is 1 sign bit and 7 fractional bits.

40.8.117 GMAC Screening Type 1 Register x Priority Queue

Name: GMAC_ST1RPQx[x=0..3]

Address: 0xF8008500

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	UDPE	DSTCE	UDPM			

23	22	21	20	19	18	17	16
UDPM							

15	14	13	12	11	10	9	8
UDPM				DSTCM			

7	6	5	4	3	2	1	0
DSTCM				–	QNB		

Screening type 1 registers are used to allocate up to 3 priority queues to received frames based on certain IP or UDP fields of incoming frames.

QNB: Queue Number (0–2)

If a match is successful, then the queue value programmed in bits 2:0 is allocated to the frame.

DSTCM: Differentiated Services or Traffic Class Match

When DS/TC match enable is set (bit 28), the DS (differentiated services) field of the received IPv4 header or TC field (traffic class) of IPv6 headers are matched against bits 11:4.

UDPM: UDP Port Match

When UDP port match enable is set (bit 29), the UDP Destination Port of the received UDP frame is matched against bits 27:12.

DSTCE: Differentiated Services or Traffic Class Match Enable

When DS/TC match enable is set (bit 28), the DS (differentiated services) field of the received IPv4 header or TC field (traffic class) of IPv6 headers are matched against bits 11:4.

UDPE: UDP Port Match Enable

When UDP port match enable is set (bit 29), the UDP Destination Port of the received UDP frame is matched against bits 27:12.

46.6.3 Master Mode

46.6.3.1 Definition

The master is the device that starts a transfer, generates a clock and stops it. This operating mode is not available if High-speed mode is selected.

46.6.3.2 Programming Master Mode

The following registers must be programmed before entering Master mode:

1. TWIHS_MMR.DADR (+ IADRSZ + IADR if a 10-bit device is addressed): The device address is used to access slave devices in Read or Write mode.
2. TWIHS_CWGR.CKDIV + CHDIV + CLDIV: Clock Waveform register
3. TWIHS_CR.SVDIS: Disables the Slave mode
4. TWIHS_CR.MSEN: Enables the Master mode

Note: If the TWIHS is already in Master mode, the device address (DADR) can be configured without disabling the Master mode.

46.6.3.3 Transfer Rate Clock Source

The TWIHS speed is defined in the TWIHS_CWGR. The TWIHS baud rate can be based either on the peripheral clock if the CKSRC bit value is ‘0’ or on a GCLK clock if the CKSRC bit value is ‘1’.

If CKSRC = 1, the baud rate is independent of the system/core clock (MCK) and thus the MCK frequency can be changed without affecting the TWIHS transfer rate.

The GCLK frequency must always be three times lower than the peripheral clock frequency.

46.6.3.4 Master Transmitter Mode

This operating mode is not available if High-speed mode is selected.

After the master initiates a START condition when writing into the Transmit Holding register (TWIHS_THR), it sends a 7-bit slave address, configured in the Master Mode register (DADR in TWIHS_MMR), to notify the slave device. The bit following the slave address indicates the transfer direction, 0 in this case (MREAD = 0 in TWIHS_MMR).

The TWIHS transfers require the slave to acknowledge each received byte. During the acknowledge clock pulse (9th pulse), the master releases the data line (HIGH), enabling the slave to pull it down in order to generate the acknowledge. If the slave does not acknowledge the byte, then the Not Acknowledge flag (NACK) is set in the TWIHS Status Register (TWIHS_SR) of the master and a STOP condition is sent. The NACK flag must be cleared by reading TWIHS_SR before the next write into TWIHS_THR. As with the other status bits, an interrupt can be generated if enabled in the Interrupt Enable register (TWIHS_IER). If the slave acknowledges the byte, the data written in the TWIHS_THR is then shifted in the internal shifter and transferred. When an acknowledge is detected, the TXRDY bit is set until a new write in the TWIHS_THR.

TXRDY is used as Transmit Ready for the DMA transmit channel.

While no new data is written in the TWIHS_THR, the serial clock line is tied low. When new data is written in the TWIHS_THR, the SCL is released and the data is sent. Setting the STOP bit in TWIHS_CR generates a STOP condition.

After a master write transfer, the serial clock line is stretched (tied low) as long as no new data is written in the TWIHS_THR or until a STOP command is performed.

To clear the TXRDY flag, first set the bit TWIHS_CR.MSDIS, then set the bit TWIHS_CR.MSEN.

See Figure 46-4, Figure 46-5, and Figure 46-6.

46.6.3.9 Using the DMA Controller (DMAC) in Master Mode

The use of the DMA significantly reduces the CPU load.

To ensure correct implementation, follow the programming sequences below:

- Data Transmit with the DMA in Master Mode

If Alternative Command mode is disabled (ACMEN bit set to ‘0’):

The DMA transfer size must be defined with the buffer size minus 1. The remaining character must be managed without DMA to ensure that the exact number of bytes are transmitted regardless of system bus latency conditions during the end of the buffer transfer period.

1. Initialize the DMA (channels, memory pointers, size - 1, etc.);
2. Configure the Master mode (DADR, CKDIV, MREAD = 0, etc.) or Slave mode.
3. Enable the DMA.
4. Wait for the DMA status flag indicating that the buffer transfer is complete.
5. Disable the DMA.
6. Wait for the TXRDY flag in TWIHS_SR.
7. Set the STOP bit in TWIHS_CR.
8. Write the last character in TWIHS_THR.
9. (Only if peripheral clock must be disabled) Wait for the TXCOMP flag to be raised in TWIHS_SR.

If Alternative Command mode is enabled (ACMEN bit set to ‘1’):

1. Initialize the transmit DMA (memory pointers, transfer size).
2. Configure the Master mode (DADR, CKDIV, etc.) and TWIHS_ACR.
3. Start the transfer by setting the DMA TXTEN bit.
4. Wait for the DMA ENDTX flag either by using the polling method or ENDTX interrupt.
5. Disable the DMA by setting the DMA TXTDIS bit.
6. (Only if peripheral clock must be disabled) Wait for the TXCOMP flag to be raised in TWIHS_SR.

- Data Receive with the DMA in Master Mode

If Alternative Command mode is disabled (ACMEN bit set to ‘0’):

The DMA transfer size must be defined with the buffer size minus 2. The two remaining characters must be managed without DMA to ensure that the exact number of bytes are received regardless of system bus latency conditions encountered during the end of buffer transfer period.

1. Initialize the DMA (channels, memory pointers, size - 2, etc.);
2. Configure the Master mode (DADR, CKDIV, MREAD = 1, etc.) or Slave mode.
3. Enable the DMA.
4. (Master Only) Write the START bit in the TWIHS_CR to start the transfer.
5. Wait for the DMA status flag indicating that the buffer transfer is complete.
6. Disable the DMA.
7. Wait for the RXRDY flag in the TWIHS_SR.
8. Set the STOP bit in TWIHS_CR.
9. Read the penultimate character in TWIHS_RHR.
10. Wait for the RXRDY flag in the TWIHS_SR.
11. Read the last character in TWIHS_RHR.
12. (Only if peripheral clock must be disabled) Wait for the TXCOMP flag to be raised in TWIHS_SR.

RXFPTEF: Receive FIFO Pointer Error Flag

0: No Receive FIFO pointer occurred

1: Receive FIFO pointer error occurred. Receiver must be reset

See Section 47.9.6.10 ”FIFO Pointer Error” for details.

49.8.11 SPI Interrupt Mask Register

Name: SPI_IMR

Address: 0xF800001C (0), 0xFC00001C (1)

Access: Read-only

31	30	29	28	27	26	25	24
RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	CMP	UNDES	TXEMPTY	NSSR
7	6	5	4	3	2	1	0
–	–	–	–	OVRES	MODF	TDRE	RDRF

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

RDRF: Receive Data Register Full Interrupt Mask

TDRE: SPI Transmit Data Register Empty Interrupt Mask

MODF: Mode Fault Error Interrupt Mask

OVRES: Overrun Error Interrupt Mask

NSSR: NSS Rising Interrupt Mask

TXEMPTY: Transmission Registers Empty Mask

UNDES: Underrun Error Interrupt Mask

CMP: Comparison Interrupt Mask

TXFEF: TXFEF Interrupt Mask

TXFFF: TXFFF Interrupt Mask

TXFTHF: TXFTHF Interrupt Mask

RXFEF: RXFEF Interrupt Mask

RXFFF: RXFFF Interrupt Mask

RXFTHF: RXFTHF Interrupt Mask

TXFPTEF: TXFPTEF Interrupt Mask

RXFPTEF: RXFPTEF Interrupt Mask

50. Quad Serial Peripheral Interface (QSPI)

50.1 Description

The Quad Serial Peripheral Interface (QSPI) is a synchronous serial data link that provides communication with external devices in Master mode.

The QSPI can be used in SPI mode to interface to serial peripherals such as ADCs, DACs, LCD controllers, CAN controllers and sensors, or in Serial Memory mode to interface to serial Flash memories.

The QSPI allows the system to execute code directly from a serial Flash memory (XIP) without code shadowing to RAM. The serial Flash memory mapping is seen in the system as other memories such as ROM, SRAM, DRAM, embedded Flash memory, etc.

With the support of the Quad SPI protocol, the QSPI allows the system to use high-performance serial Flash memories which are small and inexpensive, in place of larger and more expensive parallel Flash memories.

Note: Stacked devices with a rollover in the memory address space at each die boundary are not supported.

50.2 Embedded Characteristics

- Master SPI Interface
  - Programmable clock phase and clock polarity
  - Programmable transfer delays between consecutive transfers, between clock and data, between deactivation and activation of chip select
- SPI Mode
  - Interface to serial peripherals such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - 8-bit/16-bit/32-bit programmable data length
- Serial Memory Mode
  - Interface to serial Flash memories operating in Single-bit SPI, Dual SPI and Quad SPI
  - Interface to serial Flash Memories operating in Single Data Rate Mode
  - Supports “Execute In Place” (XIP)— code execution by the system directly from a serial Flash memory
  - Flexible instruction register for compatibility with all serial Flash memories
  - 32-bit address mode (default is 24-bit address) to support serial Flash memories larger than 128 Mbits
  - Continuous read mode
  - Scrambling/unscrambling “On-The-Fly”
- Connection to DMA Channel Capabilities Optimizes Data Transfers
  - One channel for the receiver, one channel for the transmitter
- Register Write Protection

55.7.5 PDMIC Interrupt Disable Register

Name: PDMIC_IDR

Address: 0xF801801C

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	OVRE	DRDY
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

DRDY: Data Ready Interrupt Disable

OVRE: General Overrun Error Interrupt Disable

57.5.3 SFC Interrupt Enable Register

Name: SFC_IER

Address: 0xF804C010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	ACE	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	LCHECK	–	–	PGMF	PGMC

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt

PGMC: Programming Sequence Completed Interrupt Enable

PGMF: Programming Sequence Failed Interrupt Enable

LCHECK: Live Integrity Check Error Interrupt Enable

ACE: Area Check Error Interrupt Enable

Figure 58-4: Region Descriptor

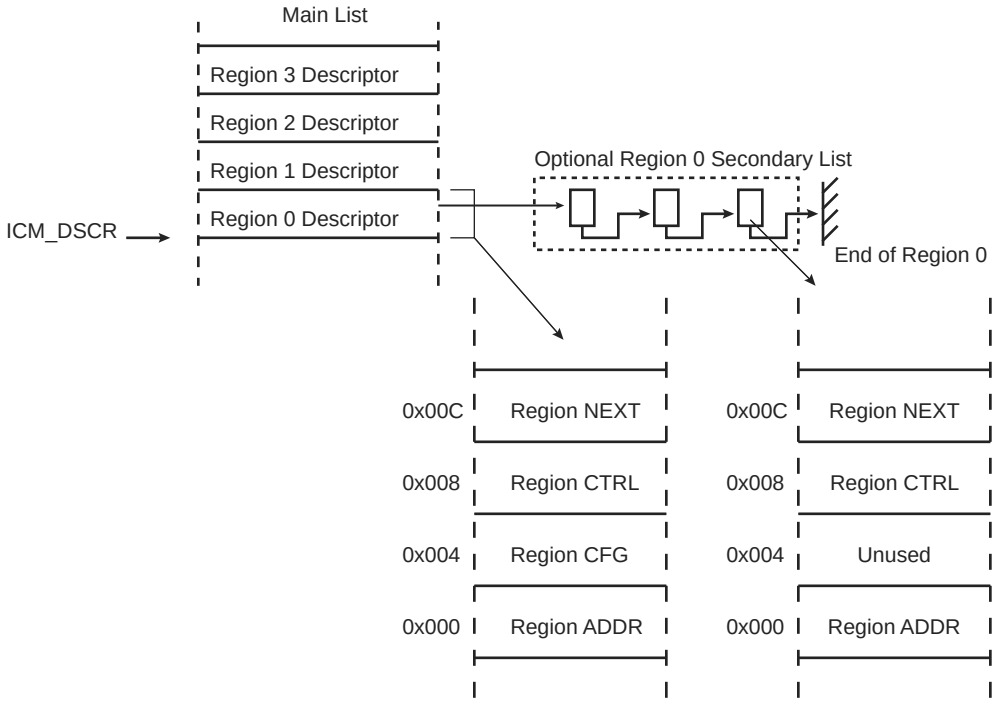


Figure 58-5 shows an example of the mandatory ICM settings to monitor three memory data blocks of the system memory (defined as two regions) with one region being not contiguous (two separate areas) and one contiguous memory area. For each region, the SHA algorithm may be independently selected (different for each region). The wrap allows continuous monitoring.

Value	Name	Description
0x2	–	Reserved
0x3	–	Reserved
0x4	CTR	Counter mode (16-bit internal counter)

Values which are not listed in the table must be considered as “reserved”.

For CBC-MAC operating mode, configure OPMOD to 0x1 (CBC) and set LOD to 1.

Note: If the OPMODE field is set to 0x4 and AAHB = 1, there is no compliance with the standard CTR mode of operation.

LOD: Last Output Data Mode

0: No effect.

After each end of encryption/decryption, the output data will be available either on the output data registers (Manual and Auto modes).

In Manual and Auto modes, the AESB_ISR.DATRDY bit is cleared when at least one of the Output Data registers is read.

1: The AESB_ISR.DATRDY bit is cleared when at least one of the Input Data Registers is written.

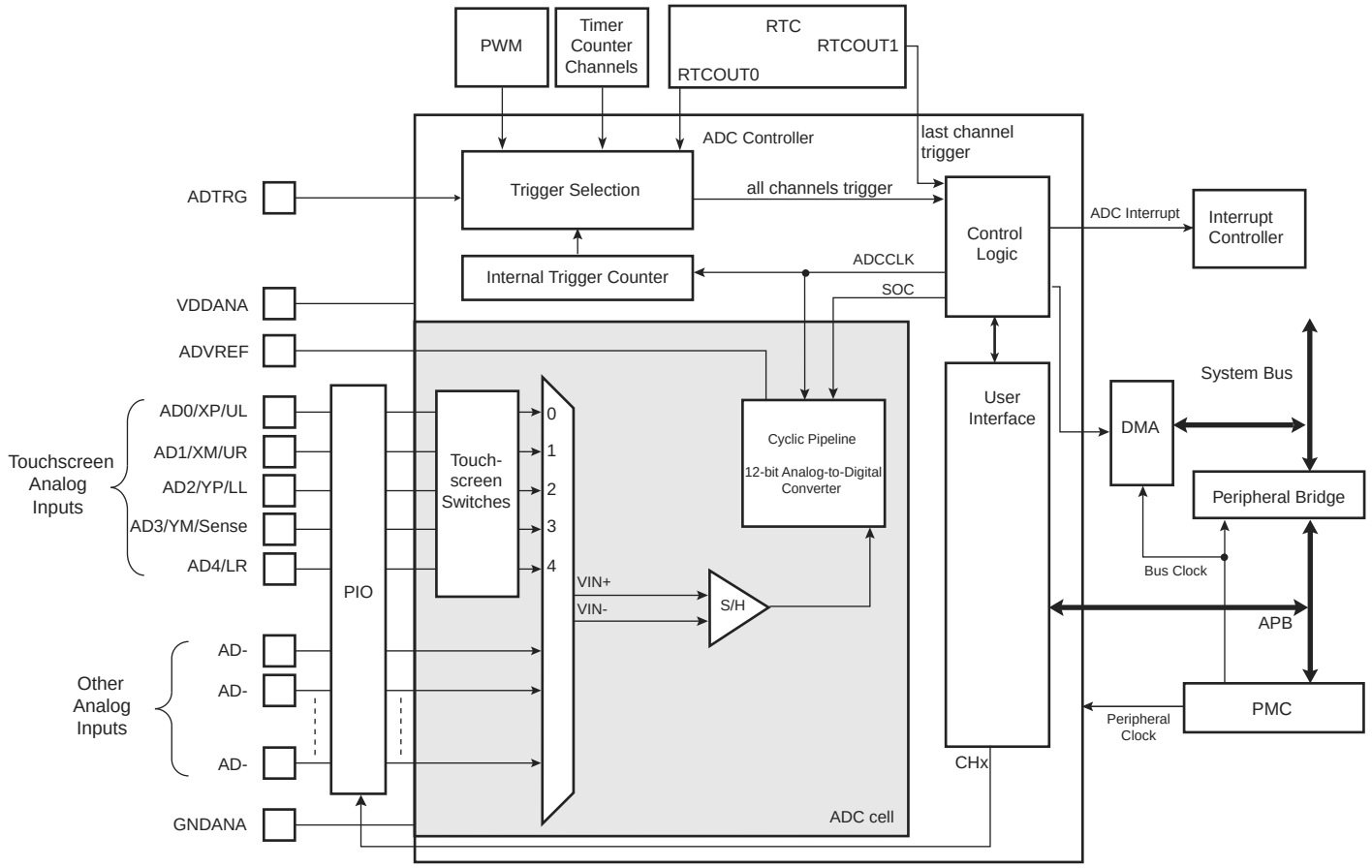
No more Output Data Register reads are necessary between consecutive encryptions/decryptions (refer to Section 59.3.4 “Last Output Data Mode”).

CKEY: Key

Value	Name	Description
0xE	PASSWD	This field must be written with 0xE the first time that AESB_MR is programmed. For subsequent programming of the AESB_MR register, any value can be written, including that of 0xE.Always reads as 0.

65.3 Block Diagram

Figure 65-1: Analog-to-Digital Converter Block Diagram



65.4 Signal Description

Table 65-1: ADC Pin Description

Pin Name	Description
VDDANA	Analog power supply
ADVREF	Reference voltage
AD0–AD11	Analog input channels
ADTRG	External trigger

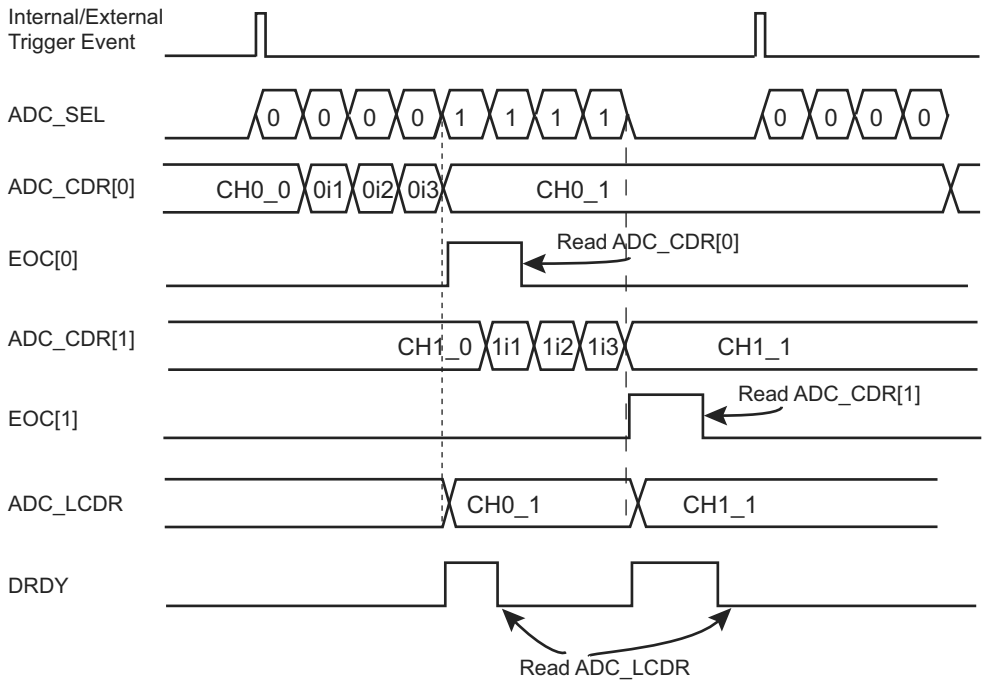
65.5 Product Dependencies

65.5.1 Power Management

The ADC Controller is not continuously clocked. The programmer must first enable the ADC Controller peripheral clock in the Power Management Controller (PMC) before using the ADC Controller. However, if the application does not require ADC operations, the ADC Controller clock can be stopped when not needed and restarted when necessary. Configuring the ADC Controller does not require the ADC Controller clock to be enabled.

Figure 65-13: Digital Averaging Function Waveforms on a Single Trigger Event, Non-interleaved

ADC_EMR.OSR = 1, ASTE = 1, ADC_CHSR[7:0] = 0xFF and ADC_MR.USEQ = 1
ADC_SEQR1 = 0x1111_0000



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

65.6.14 Automatic Error Correction

The ADC features automatic error correction of conversion results. Offset and gain error corrections are available. The correction can be enabled for each channel and correction values (offset and gain) are the same for all channels.

To enable error correction, the corresponding ECORRx bit must be set in the Channel Error Correction Register (ADC_CECR). The offset and gain values used to compensate the results are the same for all correction-enabled channels and programmed in the Correction Values Register (ADC_CVR).

The error correction for channels used with the touchscreen is available in the ADC Touchscreen Correction Values Register (ADC_TSCVR).

The ADCMODE field in ADC_EMR is used to configure a running mode of the ADC Normal mode, Offset Error mode, or Gain Error mode (see Section 65.7.16 “ADC Extended Mode Register”). ADCMODE uses 3 internal references to be measured and to extract the offset and gain error from 3 point-measurement codes. If some references already exist on the final application connected to some input channel ADx, they can be used as a replacement of the ADCMODE to generate the 2 or 3 points of calibration and used to extract the GAINCORR and OFFSETCORR.

After a reset, the running mode of the ADC is Normal mode. Offset Error mode and Gain Error mode are used to determine values of offset compensation and gain compensation, respectively, to apply to conversion results. Table 65-7 provides formulas to obtain the compensation values, with:

- OFFSETCORR—the Offset Correction value. OFFSETCORR is a signed value.
- GAINCORR—the Gain Correction value
- GCi—the intermediate Gain Compensation value
- Gs—the value 13
- ConvValue—the value converted by the ADC (as returned in ADC_LCDR or ADC_CDR)
- Resolution—the resolution used to process the conversion (either RESOLUTION, RESOLUTION+1 or RESOLUTION+2).